Manufacturer's Mixed Pallet Design Problem

Hande Yaman
Alper Şen

Department of Industrial Engineering
Bilkent University
Bilkent, Ankara 06800, Turkey
{hyaman,alpersen}@bilkent.edu.tr

Abstract

We study a problem faced by a major beverage producer. The company produces and distributes several brands to various customers from its regional distributors. For some of these brands, most customers do not have enough demand to justify full pallet shipments. Therefore, the company decided to design a number of mixed or "rainbow" pallets so that its customers can order these unpopular brands without deviating too much from what they initially need. We formally state the company's problem as determining the contents of a pre–determined number of mixed pallets so as to minimize the total inventory holding and backlogging costs of its customers over a finite horizon. We first show that the problem is NP–hard. We then formulate the problem as a mixed integer linear program, and incorporate valid inequalities to strengthen the formulation. Finally, we use company data to conduct a computational study to investigate the efficiency of the formulation and the impact of mixed pallets on customers' total costs.

Keywords Distribution, mixed pallet design, complexity, valid inequalities

1. Introduction

The main motivation behind this research is our experience with a leading Turkish beverage producer. The company dominates the Turkish market in its category with a single brand. Recently, the company introduced a number of new brands, two of them produced under license agreements with international companies. These *marginal* products are only produced and packaged in a facility in Istanbul and are shipped to five regional distribution centers in full pallets. The regional distribution centers then distribute these products to major vendors or redistributors in their own regions. These new brands, however, have not established sufficient demand in many vendors and redistributors to justify full pallet shipments (a full pallet may include as many as 1,728 units) from the regional distribution centers. For some vendors, a full pallet of a particular brand would even exceed the total demand in six months. Worrying about inventory costs and potential perishability issues, the vendors are not willing to order these brands in full pallets. If given flexibility, a vendor would dictate a particular custom pallet each time she orders and would specify the number of cases of each brand in the pallet depending on her consumption and future needs. Not surprisingly, this kind of operation is very costly and complicated for the beverage company. The company does not have the technology to design and create mixed pallets in the regional distribution centers. Workers need to break up full pallets, pick individual products and load mixed pallets manually. All of these operations take substantial amount of time and are subject to mistakes and accidents. For these reasons, the distribution division initially resisted the idea to create custom pallets per each customer order. After receiving a lot of push–back from the customers and sales department, the distribution division asked our help to design standard mixed pallets that would be created in the production facility in Istanbul. Since the product mix can vary among different customers, the idea is to come up with a sufficient number of standard mixed pallets that would enable the customers to order these marginal products without ordering too little or too much of what they initially need.

Like many aspects of business, product proliferation has a substantial impact on materials handling, especially in food and beverage industries (Modern Materials Handling 2001). As the Just–in–Time (JIT) or Efficient Consumer Response (ECR) strategies become more dominant, the retailers no longer accept bulk shipments from their suppliers. The industry is moving from shipping products in uniform pallets in truckloads once a week to shipping products in mixed pallet loads (or "rainbow pallets") delivered less–than–truckload (LTL), two or three times per week (Andel, 1998). Mixed pallets also allow the manufacturers to penetrate the small size retailer market. For example, Pennsylvania based New World Pasta offers a mixed pallet that consists of a variety of items, such as thin spaghetti, linguini, fetuccini and angel hair to attract retailers that otherwise cannot take a full pallet of the same item (Barrese, 2002).

In addition to providing purchasing and inventory efficiencies, mixed pallets are also used to display merchandise in stores. The mixed pallets that are created in manufacturing facilities or warehouses with all cartons facing out, toward the customer, are directly moved into the stores (Witt, 1995). For example, a paper products company featuring picnic supplies collected data on relative demand of each product and developed a picnic display that had the right number of each item on the pallet (Grocery Marketing 1996). Store ready mixed pallets are effectively used also for ice creams, canned and frozen goods from Campbell/Swanson and Sara Lee pies (Redman, 1996). Using these mixed pallets, retailers are able to replenish the merchandise right off the store floor replacing one pallet with another.

Despite undisputable benefits at the retailer side, offering mixed pallets may complicate the logistics. First, the manufacturer or the warehouse needs to decide how many of each product should be placed in the pallet. For the case where the warehouse is capable of designing and creating a mixed pallet per customer requirement, this decision is taken each time a customer order is placed. For the case where the warehouse does not have such capability, the manufacturer needs to design standard mixed pallets and create these mixed pallets upfront. The customers pick among these standard mixed pallets when they order. While deciding the contents of the standard mixed pallets, the manufacturer needs to consider different demand mixes of its many customers and make sure that what they order does not deviate too much from what they originally need. This is in fact the problem we consider in this research.

Even when the contents or the product mix is determined, the physical design of a mixed pallet can still be challenging because of the differences in product dimensions and weight. The design should enable efficient use of pallet space and ensure a stable load. When mixed pallets are used for store display, the physical design should also consider the sales impact. Therefore, designing mixed pallets manually could be an expensive and time consuming task. A number of software companies provide solutions that support decision making in the physical design of mixed pallets. Two such palletizing solutions are Cape Pack from Cape Systems, and TOPS Pro from Tops Engineering Corp. (Food & Drug Packaging 2000).

Creating mixed pallets physically is labor intensive and costly because of damages and mistakes. Labor safety is also at risk, as full pallets need to be depalletized and individual product cases need to be picked and loaded onto the mixed pallets. For the case of customized mixed pallets, warehouses either need to employ a vast number of operators in the loading area, or need to have the technology to automate the palletizing; former increase operating costs substantially and latter needs substantial investment. A number of automated equipment can be used to create mixed pallets. Robotic palletizers can pick random box sizes and build efficient mixed pallets. Two examples are robotics palletizing/depalletizing systems from Fanuc Robotics and FKI Logistex (Aichlmayr, 2002). Another alternative is the use of automated storage and retrieval systems (AS/RS) which could release a full pallet of product, depalletize it and use the individual cases to form mixed pallets. An example for such systems is Modular Storage System (MSS II) from Daifuku (formerly Eskay) (Beverage Industry, 2001).

Academic literature on constructing pallets mainly focused on physical arrangement of the cases in pallets in such a way that the maximum use of the pallet space is achieved. Research in this area initially is concerned with the arrangement of identical cases on a pallet, which has been termed as the "Manufacturer's Pallet Packing Problem" (Bischoff and Ratcliff, 1995). The assumption here is that the manufacturer is constructing a pallet of a single item (or different items with cases of same dimensions). The problem is a specialized version of the three dimensional cutting stock problem (Gilmore and Gomory, 1965). We refer the reader to Hifi (2004) for a review of literature and algorithms for solving the three dimensional cutting stock problem. A related problem, which has been termed as the "Distributor's Pallet Packing Problem" is concerned with the arrangement of cases of different sizes on multiple pallets. The assumption in this problem is that the distributor is constructing multiple pallets to meet a particular demand of different items packed in cases of different sizes. This problem also can be modeled as a three dimensional cutting stock problem, which is NP–hard, but for which many solution methods have been suggested (Hifi, 2004). We should note that the Manufacturer's Pallet Packing Problem needs to be solved only once and when a particular pattern of the cases is determined, the same pattern is used for all pallets. Distributor's Pallet Packing Problem, on the other hand, is a tactical problem that needs to be solved each time the distributor needs to ship items to a retailer and requires the formation of multiple pallets with different patterns. We finally note that the structures of these two problems apply to a number of different problem areas such as container loading, loading containers into ships or loading trucks.

In this study, we are not concerned with the physical arrangement of the cases in pallets, as all cases in our application are of identical dimensions regardless of the beverage they contain. The arrangement of the cases in the pallet are also pre–determined, i.e., each pallet has a fixed number of rows and each row can take a fixed number of cases. However, different from the Manufacturer's and Distributor's Pallet Packing Problems, our problem is to determine the number of cases of each product to be loaded onto a pre–determined number of standard mixed pallets. Unlike the Distributor's Pallet Packing Problem, our problem is not a tactical one, as we are not designing mixed pallets per customer order. The mixed pallets are standard and the customers are to choose among the available mixed pallets when they order. We assume that the customers determine the number of mixed and full pallets to order in each period so as to minimize their inventory holding and backlogging costs over a finite horizon.

The remainder of the paper is organized as follows. In Section 2, we formally introduce our problem and show that it is NP–hard. In Section 3, we formulate the problem as a mixed integer linear programming problem and derive valid inequalities that strengthen the formulation. In Section 4, we conduct a numerical study to test the efficiency of the formulation and the valid inequalities and to assess the impact of mixed pallets on the customer inventory holding and backlogging costs. We conclude the paper and state the avenues for future research in Section 5.

2. Problem Definition and Complexity

We are given a set of customers C and a set of products N. Let $T = \{1, 2, \dots, \tau\}$ be the set of periods. Each customer c has a demand of $d_{cit} \geq 0$ for product i in period t. Products are of identical dimensions and are sold in pallets. Each pallet has Q_1 units of capacity (rows). In each row, there are Q_2 units of a product. There is a pre–determined set of potential mixed pallet designs P (later in Proposition 1, we determine the maximum cardinality of this set). Pallet design j in set P has q_{ij} rows of product i and $\sum_{i \in N} q_{ij} = Q_1$ for all $j \in P$. In addition, the manufacturer offers full pallets for each product i, which consists of $Q_1 Q_2$ units of product i.

Retailers have linear inventory holding and backlogging costs. The cost of holding one unit of inventory for product i at the end of period t for customer c is h_{cit}. Likewise, the cost of backlogging one unit of demand for product i at the end of period t for customer c is π_{cit}. No backlogging is permitted at the end of period τ (i.e., all demands should be satisfied by the end of τ).

Given a set of available mixed pallet designs, each customer's problem is to determine the number of full pallets from each product and the number of mixed pallets from each design to buy in each period so as to minimize its own total inventory holding and backlogging costs in periods $1, 2, \dots, \tau$. Assuming that each customer is making its decision optimally, the manufacturer's problem is to select at most m mixed pallet designs from set P so as to minimize the sum of customers' inventory holding and backlogging costs in periods $1, 2, \dots, \tau$. We call the manufacturer's problem *mixed pallet design problem.*

In Figure 1, we explain the problem using a simple example. In this example, there are two products ($|N| = 2$), two customers ($|C| = 2$) and a single period ($\tau = 1$). Pallets have six rows ($Q_1 = 6$) and each row contains one unit of a product ($Q_2 = 1$). There are potentially five different mixed pallet designs ($|P| = 5$), and mixed pallet i contains i rows of product 2 and $6 - i$ rows of product 1. Customer 1 has $(38, 40)$ units of demand for product 1 and product 2. Customer 2 has $(22, 13)$ units of demand for product 1 and product 2. Assume that $h_{ci} = 1$ for all $c \in C$ and $i \in N$. If only full pallets are used, these customers have to purchase $(42, 42)$ and $(24, 18)$ units of product 1 and product 2, incuring a total cost of 13. The problem of the manufacturer is to find the specific mixed pallet designs to be used given a maximum number of mixed pallet designs so that the total of inventory holding and backlogging costs are minimized. For example, using a single standard mixed pallet (design 4), these customers will be able to purchase $(38, 40)$ and $(22, 14)$, incuring a total cost of 1.

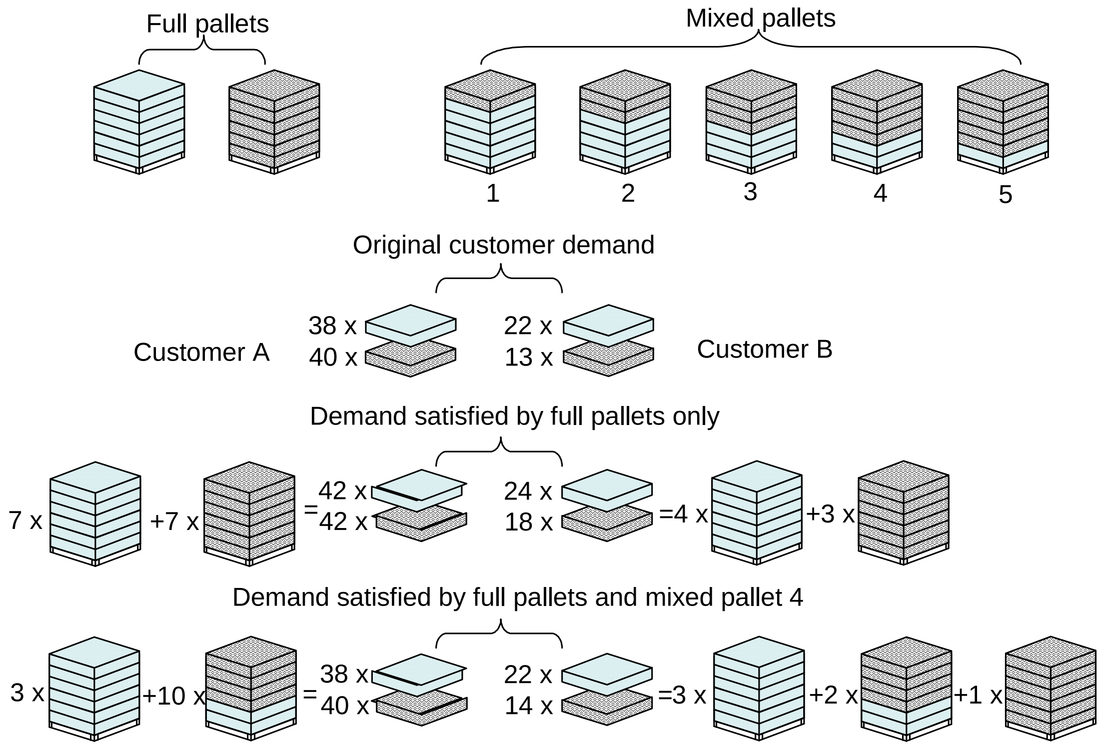

Figure 1: Example for the use of mixed pallets

We next show that the *mixed pallet design problem* is NP–hard. The proof in fact shows that even a simple one customer, two product, one period instance of the general mixed pallet design problem is NP–hard.

Theorem 1 *The mixed pallet design problem is NP–hard.*

Proof. Clearly, the decision version of the mixed pallet design problem is in class NP. Consider the integer knapsack problem. Given a set U, a size $s_u \in \mathbb{Z}_+$ and a value $v_u \in \mathbb{Z}_+$ for each $u \in U$ and positive integers B and K, does there exist $c_u \in \mathbb{Z}_+$ for each $u \in U$, such that $\sum_{u \in U} s_u c_u \leq B$ and $\sum_{u \in U} v_u c_u \geq K$? This problem is NP–complete even when $s_u = v_u$ for all $u \in U$ (see Garey and Johnson (1979), problem [MP10]).

Consider an instance of the integer knapsack problem where $s_u = v_u$ for all $u \in U$. We reduce this to an instance of the decision version of the mixed pallet design problem. Suppose that there is one customer, one period and two products. Let $Q_2 = 1$ and $Q_1 = \max_{u \in U} s_u$. We take $P = U$, $m = |U|$, $q_{1u} = s_u$ and $q_{2u} = Q_1 - s_u$ for all $u \in U$. Hence every item u of the knapsack problem corresponds to a pallet design u, which has s_u units of product 1 and $Q_1 - s_u$ units of product 2. The demand of the customer for product 1 is K and for product 2 is 0. The customer should decide how many pallets to buy of each type in order to satisfy the demand. Let $c_u \in \mathbb{Z}_+$ be the number of pallets of type $u \in U$ that the customer buys. Since the customer has to satisfy its demand, we should have $\sum_{u \in U} s_u c_u \geq K$ and there is no possibility of backlogging. The inventory cost for product 1 is 1 and for product 2 is 0. Since the inventory cost for product 2 is 0, the total cost for the customer is equal to the inventory holding cost for product 1, i.e., $\sum_{u \in U} s_u c_u - K$. So there exists a decision for the customer with cost less than or equal to $B - K$ if and only if there exists a solution to the integer knapsack problem with $\sum_{u \in U} s_u c_u \geq K$ and $\sum_{u \in U} s_u c_u \leq B$. $\square$

Now, we discuss various assumptions regarding our problem. First, we assume that each customer is of equal importance to the manufacturer. This can be easily relaxed by incorporating weights to each customer. We also assume that the choice of m is external considering various factors including complexity in operations and impact on pallet inventories. Clearly, larger m values provide better service to the customers, as the company is better able to match the demand of each customer.

We also assume that the manufacturer has a pre–determined set P of potential mixed pallet designs to choose from. In the event that the manufacturer does not have a pre–determined set, we work with the set of all possible mixed pallet designs. Next, we derive the cardinality of the set of all possible mixed pallet designs.

Let $N(n, Q_1)$ be the number of possible mixed pallet designs (including the designs that has only one color, i.e., full pallets) if there are n products and Q_1 rows in a mixed pallet. By definition $N(1, Q_1) = 1$, for any Q_1. In order to calculate the number of designs for general $n > 1$, we use the following proposition.

Proposition 1

$$N(n, r) = \sum_{x=1}^{r} N(n-1, x) + 1$$

Proof. Assume that at every stage, we are deciding on the number of rows to be used for a single product. Assume that we start a particular stage with r rows and n products. If the next product is used in $r - x$ rows ($1 \leq x \leq r$) at that stage, then the remaining problem is one with $n-1$ products and x rows. We have an additional 1 in the equation since the product can be assigned to all remaining r rows which corresponds to a single design, regardless of how many products are left. $\square$

In order to find $N(n, Q_1)$ for any n, we need to calculate the sum of power series recursively. The results for $n \leq 4$ are given below:

$$
\begin{aligned}
N(1,Q_1) &= 1 \\
N(2,Q_1) &= \sum_{x=1}^{Q_1} N(1,x)+1 = \sum_{x=1}^{Q_1} 1+1 = Q_1+1 \\
N(3,Q_1) &= \sum_{x=1}^{Q_1} N(2,x)+1 = \sum_{x=1}^{Q_1} (x+1)+1 = \frac{Q_1^2+3Q_1+2}{2} \\
N(4,Q_1) &= \sum_{x=1}^{Q_1} N(3,x)+1 = \sum_{x=1}^{Q_1} (\frac{x^2+3x+2}{2})+1 = \frac{Q_1^3+6Q_1^2+11Q_1+6}{6}
\end{aligned}
$$

We note that $N(n,Q_1)$ also includes the full pallets, therefore $|P| = N(n,Q_1) - n$.

3. Problem Formulation

In this section, we provide a mathematical programming formulation of the mixed pallet design problem. We define the following decision variables for our formulation.

$$
\begin{aligned}
p_j &: \begin{cases} 1, & \text{if pallet design } j \text{ is offered} \\ 0, & \text{otherwise} \end{cases} \\
y_{cjt} &: \text{number of pallets of type } j \text{ that customer } c \text{ buys in period } t \\
f_{cit} &: \text{number of full pallets of product type } i \text{ that customer } c \text{ buys in period } t \\
I_{cit} &: \text{amount of inventory of product } i \text{ that customer } c \text{ has at the end of period } t \\
B_{cit} &: \text{amount of product } i \text{ that is backlogged at the end of period } t \text{ for customer } c
\end{aligned}
$$

In addition, let P_c denote the set of mixed pallets that customer c can buy. Let M be a very large number. Using the variables above, it is possible to obtain a linear mixed integer programming formulation. This formulation, called *MPD*, is as follows:

$$
(MPD) \qquad \min \sum_{c\in C} \sum_{i\in N} \sum_{t\in T} (\pi_{cit} B_{cit} + h_{cit} I_{cit}) \tag{1}
$$

$$
\text{s.t.} \quad \sum_{j\in P} p_j \le m \tag{2}
$$

$$
I_{cit-1} - B_{cit-1} + Q_1 Q_2 f_{cit} + \sum_{j\in P_c} Q_2 q_{ij} y_{cjt} = d_{cit} + I_{cit} - B_{cit} \qquad \forall c \in C, i \in N, t \in T \tag{3}
$$

$$
y_{cjt} \le M p_j \qquad \forall c \in C, j \in P_c, t \in T \tag{4}
$$

$$
I_{ci0} = B_{ci0} = B_{ci\tau} = 0 \qquad \forall c \in C, i \in N \tag{5}
$$

$$
I_{cit}, B_{cit} \ge 0 \qquad \forall c \in C, i \in N, t \in T \tag{6}
$$

$$f_{cit} \geq 0 \text{ and integer} \qquad \forall c \in C, i \in N, t \in T \tag{7}$$

$$y_{cjt} \geq 0 \text{ and integer} \qquad \forall c \in C, j \in P_c, t \in T \tag{8}$$

$$p_j \in \{0,1\} \qquad \forall j \in P. \tag{9}$$

Constraint (2) limits the number of mixed pallet designs to be used to m. Constraints (3) are the balance equations where the number of product type i that customer c receives in period t is $Q_1 Q_2 f_{cit} + \sum_{j \in P_c} Q_2 q_{ij} y_{cjt}$. Constraints (4) forbid any customer to buy a certain pallet design if this design is not offered. Constraints (5) impose beginning and ending conditions. Constraints (6)–(9) state the types of decision variables. Objective function (1) is the sum of inventory holding and backlogging costs over all periods. The aim is to minimize this total cost.

The advantage of this formulation is that constraints forbidding some mixed pallet designs for some or all customers can be incorporated very easily. However, it has the disadvantage that, if $|P|$ is large, then the number of variables is large. For the application we consider, since the number of products is small, $|P|$ is relatively small and LP relaxations are solved efficiently.

Another concern is the strength of the formulation. The above formulation has a very weak LP relaxation. Indeed, solution of the LP relaxation only gives a trivial bound.

Proposition 2 *The optimal value of the LP relaxation of MPD is equal to zero.*

Proof. Clearly, the optimal value of the LP relaxation is nonnegative. Consider the solution given by $p_j = 0$ for all $j \in P$, $y_{cjt} = 0$ for all $c \in C$, $j \in P_c$ and $t \in T$, $f_{cit} = \frac{d_{cit}}{Q_1 Q_2}$ for all $c \in C$, $i \in N$ and $t \in T$, $I_{cit} = B_{cit} = 0$ for all $c \in C$, $i \in N$ and $t \in T$. As this solution is feasible for the LP relaxation and it has objective function value of zero, it is optimal. $\square$

It is important to improve this lower bound to be able to solve the problem to optimality. In the remainder of this section, we try to improve this formulation by choosing a good value of M and by adding valid inequalities.

3.1 Choice of M and Aggregation of Constraints (4)

The number of constraints (4) can be large since there is a constraint per pallet design, customer and period. It may be important to decrease the number of these constraints to improve the solution time. A common technique is aggregation. These constraints can be aggregated in the following ways:

$$\sum_{c \in C: j \in P_c} y_{cjt} \leq M p_j \qquad \forall j \in P, t \in T \tag{10}$$

or

$$\sum_{t \in T} y_{cjt} \leq M p_j \qquad \forall c \in C, j \in P_c \tag{11}$$

or

$$\sum_{c \in C: j \in P_c} \sum_{t \in T} y_{cjt} \leq M p_j \qquad \forall j \in P. \tag{12}$$

Each aggregation leads to a valid formulation of *MPD*. The relative strengths of these aggregated inequalities depend on the value of M. For the same M, inequality (12) is stronger than inequalities (10) and (11), and they are stronger than inequality (4). But it may be possible to choose better M values for disaggregated inequalities.

Given the pallet designs to be used, all customers behave independently. So if inequality (12) is valid with M, then there exist $M_1, M_2, \ldots, M_{|C|}$ such that $\sum_{t \in T} y_{cjt} \leq M_c\, p_j$ is valid for each $c \in C$ such that $j \in P_c$ and $\sum_{c \in C} M_c = M$. So inequality (12) cannot dominate inequalities (11) if the upper bounds are tight.

Next, we present tight upper bounds to use in inequalities (11).

Proposition 3 *There exists an optimal solution which satisfies*

$$\sum_{t \in T} y_{cjt} \leq \max_{i \in N: q_{ij} > 0} \left\lceil \frac{\sum_{t \in T} d_{cit}}{q_{ij} Q_2} \right\rceil p_j \tag{13}$$

for all $c \in C$ *and* $j \in P_c$.

Proof. As backlogging is allowed, a customer can buy all the demand of a product in any period. By feasibility, we can say that in a given period, customer c can buy $\max_{i \in N: q_{ij} > 0} \left\lceil \frac{{}_{t \in T} d_{cit}}{q_{ij} Q_2} \right\rceil$ pallets of type j. There exists a feasible solution where the bound is attained. But it is also true that, by optimality, customer c does not buy more than this quantity over all periods. □

In our formulation, we replace constraints (4) with inequalities (13) for all $c \in C$ and $j \in P_c$. This way, we decrease the number of constraints (4) by an order of $|T|$ without sacrificing from the strength of the formulation. In fact, the LP bound is still zero as the solution given in the proof of Proposition 2 is still feasible. To improve the LP bound, we derive valid inequalities.

3.2 Valid Inequalities

In this section, we derive valid inequalities using relaxations of the problem and mixed integer rounding (Marchand and Wolsey 2001, Nemhauser and Wolsey 1988). Similar ideas have often been used to solve different production planning problems (see e.g. Belvaux and Wolsey 2000, Miller and Wolsey 2003, Pochet and Wolsey 2006).

The valid inequalities we obtain are based on the following idea. Consider customer $c \in C$ and a subset of mixed pallets $P' \subseteq P_c$. If none of the mixed pallets in the set $P_c \setminus P'$ is offered, then customer c has to satisfy his/her demand using full pallets and mixed pallets of the set P'.

Proposition 4 *Let $c \in C$, $i \in N$, $t_1, t_2 \in T$ such that $t_1 \leq t_2$, $\alpha \in \mathbb{Z}_+$ and $D_{ci}(t_1,t_2,\alpha) = \sum_{t=t_1}^{t_2} d_{cit}/\alpha$ with $D_{ci}(t_1,t_2,\alpha)$ not integer and $P' \subseteq P_c$. The inequality*

$$\sum_{t=t_1}^{t_2}\left(\min\left\{\left\lceil\frac{Q_1Q_2}{\alpha}\right\rceil, \lceil D_{ci}(t_1,t_2,\alpha)\rceil\right\}f_{cit} + \sum_{j\in P'}\min\left\{\left\lceil\frac{q_{ij}Q_2}{\alpha}\right\rceil, \lceil D_{ci}(t_1,t_2,\alpha)\rceil\right\}y_{cjt}\right)$$
$$+\frac{I_{cit_1-1}+B_{cit_2}}{\alpha(D_{ci}(t_1,t_2,\alpha)-\lfloor D_{ci}(t_1,t_2,\alpha)\rfloor)} \geq \lceil D_{ci}(t_1,t_2,\alpha)\rceil(1-\sum_{j\in P_c\setminus P':q_{ij}>0} p_j) \qquad (14)$$

is a valid inequality.

Proof. If $\sum_{j\in P_c\setminus P':q_{ij}>0} p_j \geq 1$, the right hand side of inequality (14) is nonpositive. If $\sum_{j\in P_c\setminus P':q_{ij}>0} p_j = 0$, then we need to prove that the left hand side should be at least $\lceil D_{ci}(t_1,t_2,\alpha)\rceil$. Summing inequality (3) over $t = t_1, \ldots, t_2$ yields

$$I_{cit_1-1} - B_{cit_1-1} + \sum_{t=t_1}^{t_2}\left(Q_1Q_2f_{cit} + \sum_{j\in P_c} Q_2q_{ij}y_{cjt}\right) = \sum_{t=t_1}^{t_2} d_{cit} + I_{cit_2} - B_{cit_2}.$$

Since B_{cit_1-1} and I_{cit_2} are nonnegative and $p_j = 0$ for all $j \in P_c \setminus P'$ such that $q_{ij} > 0$, we have

$$\sum_{t=t_1}^{t_2}\left(Q_1Q_2f_{cit} + \sum_{j\in P'} q_{ij}Q_2y_{cjt}\right) + I_{cit_1-1} + B_{cit_2} \geq \sum_{t=t_1}^{t_2} d_{cit}$$

which implies

$$\sum_{t=t_1}^{t_2}\left(\left\lceil\frac{Q_1Q_2}{\alpha}\right\rceil f_{cit} + \sum_{j\in P'}\left\lceil\frac{q_{ij}Q_2}{\alpha}\right\rceil y_{cjt}\right) + \frac{I_{cit_1-1}+B_{cit_2}}{\alpha} \geq D_{ci}(t_1,t_2,\alpha).$$

Now the mixed integer rounding inequality is

$$\sum_{t=t_1}^{t_2}\left(\left\lceil\frac{Q_1Q_2}{\alpha}\right\rceil f_{cit}+\sum_{j\in P'}\left\lceil\frac{q_{ij}Q_2}{\alpha}\right\rceil y_{cjt}\right)+\frac{I_{cit_1-1}+B_{cit_2}}{\alpha(D_{ci}(t_1,t_2,\alpha)-\lfloor D_{ci}(t_1,t_2,\alpha)\rfloor)}\geq\lceil D_{ci}(t_1,t_2,\alpha)\rceil$$

and is valid. Finally, using the nonnegativity of $\frac{I_{cit_1-1}+B_{cit_2}}{\alpha(D_{ci}(t_1,t_2,\alpha)-\lfloor D_{ci}(t_1,t_2,\alpha)\rfloor)}$, we can apply coefficient reduction and obtain

$$\sum_{t=t_1}^{t_2}\left(\min\left\{\left\lceil\frac{Q_1Q_2}{\alpha}\right\rceil,\lceil D_{ci}(t_1,t_2,\alpha)\rceil\right\}f_{cit}+\sum_{j\in P'}\min\left\{\left\lceil\frac{q_{ij}Q_2}{\alpha}\right\rceil,\lceil D_{ci}(t_1,t_2,\alpha)\rceil\right\}y_{cjt}\right)$$
$$+\frac{I_{cit_1-1}+B_{cit_2}}{\alpha(D_{ci}(t_1,t_2,\alpha)-\lfloor D_{ci}(t_1,t_2,\alpha)\rfloor)}\geq\lceil D_{ci}(t_1,t_2,\alpha)\rceil.$$

This proves that inequality (14) is also satisfied when $\sum_{j\in P_c\setminus P':q_{ij}>0}p_j=0$. $\square$

Further valid inequalities can be generated using some special cases of inequalities (14). First consider the case with $t_1=1$, $t_2=\tau$, $\alpha=Q_2$ and $P^{'}=P_c$. In this case, inequality (14) simplifies to

$$\sum_{t\in T}\left(\min\{Q_1,D^{'}_{ci}\}f_{cit}+\sum_{j\in P_c}\min\{q_{ij},D^{'}_{ci}\}y_{cjt}\right)\geq D^{'}_{ci}\tag{15}$$

where $D^{'}_{ci}=\lceil D_{ci}(1,\tau,Q_2)\rceil$. Let $F=\sum_{t\in T}f_{cit}$, $Y_j=\sum_{t\in T}y_{cjt}$ and $D=D^{'}_{ci}$. We can rewrite inequality (15) as

$$\sum_{l=1}^{Q_1-1}\sum_{j\in P_c:q_{ij}=l}\min\{l,D\}Y_j+\min\{Q_1,D\}F\geq D.$$

Now, let $a_l=\sum_{j\in P_c:q_{ij}=l}Y_j$ for $l=1,2,\ldots,Q_1-1$ and $a_{Q_1}=F$. Then the above inequality simplifies to

$$\sum_{l=1}^{Q_1}\min\{l,D\}a_l\geq D$$

where a_l is a nonnegative integer for $l=1,2,\ldots,Q_1$. This is a knapsack cover inequality. See Mazur (1999) and Yaman (2005) for polyhedral properties of the integer knapsack cover polyhedron and Pochet and Wolsey (1995) for the special case where the coefficients of a_l's are integer multiples of each other.

Here we use the lifted rounding inequalities given in Yaman (2005). For $k\in\mathbb{Z}_{++}$ and $a\in\mathbb{R}$, define $u_k(a)=a-\lfloor\frac{a}{k}\rfloor k$. For $k\in\{1,\ldots,Q_1\}$, the inequality

$$\sum_{l=1}^{Q_1}\min\left\{u_k(D)\left\lceil\frac{D}{k}\right\rceil,\left(u_k(D)\left\lfloor\frac{l}{k}\right\rfloor+\min\{u_k(l),u_k(D)\}\right)\right\}a_l\geq u_k(D)\left\lceil\frac{D}{k}\right\rceil$$

is valid when $u_k(D)>0$.

The equivalent inequality for *MPD* is given in the following proposition.

Proposition 5 *For $c \in C$, $i \in N$ and $k \in \{1, \ldots, Q_1\}$ with $u_k(D'_{ci}) > 0$, inequality*

$$\sum_{t \in T} \left(\min\left\{ u_k(D'_{ci}) \left\lceil \frac{D'_{ci}}{k} \right\rceil, \left(u_k(D'_{ci}) \left\lfloor \frac{Q_1}{k} \right\rfloor + \min\{u_k(Q_1), u_k(D'_{ci})\} \right) \right\} f_{cit} \right.$$
$$\left. + \sum_{j \in P_c : q_{ij} > 0} \min\left\{ u_k(D'_{ci}) \left\lceil \frac{D'_{ci}}{k} \right\rceil, \left(u_k(D'_{ci}) \left\lfloor \frac{q_{ij}}{k} \right\rfloor + \min\{u_k(q_{ij}), u_k(D'_{ci})\} \right) \right\} y_{cjt} \right)$$
$$\geq u_k(D'_{ci}) \left\lceil \frac{D'_{ci}}{k} \right\rceil. \tag{16}$$

is valid.

Note that the optimal solution of the LP relaxation does not necessarily satisfy these inequalities. If there exists $c \in C$ and $i \in N$ such that $\lceil \frac{D'_{ci}}{Q_1} \rceil > \sum_{t \in T} \frac{d_{cit}}{Q_1 Q_2}$, then the fractional solution of the LP relaxation given in the proof of Proposition 2 is cut off by inequality (16) for $k = Q_1$. In the other case, this solution is integer and so is optimal for *MPD*.

Another special case that we consider is the following: Let $\alpha = Q_2$, $k \in \{1, \ldots, Q_1\}$ such that $k \leq \lceil D_{ci}(t_1, t_2, Q_2) \rceil$ and $P' = \{j \in P_c : q_{ij} = k\}$. In this case, inequality (14) simplifies to

$$\sum_{t=t_1}^{t_2} \left(\min\{Q_1, \lceil D_{ci}(t_1, t_2, Q_2) \rceil\} f_{cit} + \sum_{j \in P_c : q_{ij} = k} k y_{cjt} \right) + \frac{I_{cit_1-1} + B_{cit_2}}{Q_2(D_{ci}(t_1, t_2, Q_2) - \lfloor D_{ci}(t_1, t_2, Q_2) \rfloor)}$$
$$\geq \lceil D_{ci}(t_1, t_2, Q_2) \rceil (1 - \sum_{j \in P_c : 0 < q_{ij}, q_{ij} \neq k} p_j). \tag{17}$$

Proposition 6 *For $c \in C$, $i \in N$, $t_1, t_2 \in T$ such that $t_1 \leq t_2$ and $k \in \{1, \ldots, Q_1\}$ such that $k \leq \lceil D_{ci}(t_1, t_2, Q_2) \rceil$, let $\omega = \frac{\lceil D_{ci}(t_1, t_2, Q_2) \rceil}{k} - \lfloor \frac{\lceil D_{ci}(t_1, t_2, Q_2) \rceil}{k} \rfloor$. If $\frac{\lceil D_{ci}(t_1, t_2, Q_2) \rceil}{k}$ is not integer, inequality*

$$\sum_{t=t_1}^{t_2} \left(\min\left\{ \left\lceil \frac{Q_1}{k} \right\rceil, \left\lceil \frac{\lceil D_{ci}(t_1, t_2, Q_2) \rceil}{k} \right\rceil \right\} f_{cit} + \sum_{j \in P_c : q_{ij} = k} y_{cjt} \right)$$
$$+ \frac{I_{cit_1-1} + B_{cit_2}}{\omega Q_2(D_{ci}(t_1, t_2, Q_2) - \lfloor D_{ci}(t_1, t_2, Q_2) \rfloor)} \geq \left\lceil \frac{\lceil D_{ci}(t_1, t_2, Q_2) \rceil}{k} \right\rceil (1 - \sum_{j \in P_c : 0 < q_{ij}, q_{ij} \neq k} p_j). \tag{18}$$

is a valid inequality.

Proof. The Mixed Integer Rounding inequality for (17) divided by k is (18). □

4. Computational Results

In this section, we report the outcomes of two experiments. In the first experiment, we want to see if the valid inequalities help in solving the *MPD*. In the second experiment, the aim is to see the effect of using mixed pallets on the total cost.

In both experiments, we use a dataset provided by the beverage producer that was discussed in Section 1. In this dataset, we have a total of 3 products and 7 customers. The customer demand data is taken from a quarterly (with monthly buckets) sales plan agreed upon by the customers. The demand data is in cases which consist of 24 units of 50cl beverages. The maximum monthly demand for any product for any customer is 2408 cases. The minimum monthly demand is 0 cases. The average is 237.82 cases. The beverage company uses pallets with six rows ($Q_1 = 6$) and each row can take 12 cases of beverages ($Q_2 = 12$). Inventory holding cost per case per month is calculated by multiplying the sales price of each brand with the average monthly interest rate of 1 %. The inventory holding cost per case per month for three products are 0.7, 0.6 and 0.625. Backlogging cost per case per month is taken as 3.5, 3, and 3.625 for these products. The set P includes all possible mixed pallet designs. A customer does not buy a mixed pallet if the total demand of the customer for any product in the pallet is zero, i.e. $P_c = \{j \in P : \sum_{t \in T} d_{cit} > 0 \; \forall i \in N : q_{ij} > 0\}$ for each customer c.

The computation is carried out on a personal computer with a 1.6 GHz Pentium M processor and 512 MB RAM. We use ILOG OPL 4.2.0.1/CPLEX 10.0.0 with its default settings except the branching priority. In branching we give priority to p_j variables. There are two reasons for this: unlike other integer variables of the formulation, p_j variables can take only two values and when they are fixed, it is possible to fix many other variables.

In our first experiment, we use twelve instances from the dataset provided by the beverage producer as well as four random instances (available at http://www.bilkent.edu.tr/~alpersen/Mixed_Pallet). We want to see the use of adding valid inequalities (14), (16) and (18). To see the effect of each family of valid inequalities, we do the following experiment. We solve each problem instance first without any valid inequalities and then with the family of valid inequalities that we test and compute the improvement in percentage root gap (the percentage root gap is equal to $\frac{opt-root}{opt} * 100$ where opt is the optimal value and $root$ is the lower bound before branching), number of nodes and cpu time. In Table 1, we report the results without valid inequalities. For every instance, we report the name of the instance (the name starts with “e” for the instances provided by the company and with “r” for the random instances, followed by $|C|$, $|N|$, m and τ), the number of rows and the number of columns of the integer program after it is reduced by the presolve function of the solver, the optimal

problem name	no. of rows	no of columns	opt. value	% root gap	cpu (in sec.s)	no. of nodes
e, 3, 2, 1, 3	34	98	271.9	62.73	5.51	11251
e, 4, 2, 1, 3	45	129	359.8	68.11	196.53	374018
e, 5, 2, 1, 3	56	160	437.5	64.84	251.51	446624
e, 6, 2, 1, 3	67	191	499.5	65.57	1532.42	2376055
e, 7, 2, 1, 3	70	199	530.3	61.79	941.24	1411451
e, 3, 2, 2, 3	34	98	226.3	58.82	59.62	149236
e, 4, 2, 2, 3	45	129	311.8	65.78	19528.23	35807408
e, 3, 3, 1, 3	80	254	321.9	60.32	12.07	7098
e, 4, 3, 1, 3	113	350	405.45	61.44	20.43	11785
e, 5, 3, 1, 3	147	449	549.95	58.21	138.09	94986
e, 6, 3, 1, 3	158	480	659.95	62.03	288.21	199684
e, 7, 3, 1, 3	161	488	690.75	59.77	323.27	231074
r, 5, 2, 1, 3	48	137	386	56.42	537.72	1052917
r, 3, 3, 1, 3	80	254	559.9	73.78	2204.69	3183175
r, 4, 2, 2, 3	29	83	346.7	27.58	1415.32	3081664
r, 3, 2, 1, 4	31	100	415	46.37	256.15	511755

Table 1: The results without valid inequalities

value, the percentage root gap, the cpu time and the number of nodes in the branch and cut tree. Remark that even though the LP bound is always zero, as the solver generates its own cuts, the percentage root gap is different from 100%.

We first test the use of inequalities (14). We add these inequalities for all customers $c \in C$, all products $i \in N$ and for all possible choices of t_1 and t_2. There remains the choice of α and P'. Here, we consider three classes. The first class corresponds to the choice $\alpha = Q_2$ and $P' = \emptyset$. In the second class, we take $\alpha = Q_2$ and $P' = P_c$ for all $c \in C$. Finally, in the third class, we take for every $k \in \{1, \ldots, Q_1\}$ such that $k \leq \sum_{t=t_1}^{t_2} \frac{d_{cit}}{Q_2}$, $\alpha = kQ_2$ and $P' = \{j \in P_c : q_{ij} = k\}$ for $c \in C$. The results are reported in Tables 2, 3, and 4. For every instance and every class of inequalities, we report the number of rows and the percentage improvements in the percentage root gap, the cpu time and the number of nodes when we add this class of inequalities.

We observe that the first class of inequalities are not useful in general to decrease the cpu time and the number of nodes. On the average, there is an increase of 34.66% in cpu time and 25.63% in the number of nodes. The second class of inequalities is useful except for two instances where the difference is not so extreme. On the average, this class improves the percentage root gap by 4.84%, the cpu time by 34% and the number of nodes by 40.36%. The third class of inequalities is useful for all instances except for two. On the average, this class improves the percentage root gap by 0.66%, the cpu time by 26.25% and the number of nodes by 53.50%.

problem name	no. of rows	% imp. in % root gap	% imp. in cpu	% imp. in nodes
e, 3, 2, 1, 3	40	-0.66	-63.09	-51.92
e, 4, 2, 1, 3	54	1.31	-16.10	-10.93
e, 5, 2, 1, 3	66	-0.11	-69.10	-71.05
e, 6, 2, 1, 3	79	0.20	-52.72	-51.40
e, 7, 2, 1, 3	88	0.92	-90.81	-84.68
e, 3, 2, 2, 3	66	-1.25	-16.53	3.06
e, 4, 2, 2, 3	86	-0.52	-33.41	-20.85
e, 3, 3, 1, 3	121	7.68	13.61	13.93
e, 4, 3, 1, 3	169	14.47	30.54	37.45
e, 5, 3, 1, 3	221	6.52	-20.64	-12.55
e, 6, 3, 1, 3	243	12.56	19.19	28.43
e, 7, 3, 1, 3	252	14.01	39.54	41.25
r, 5, 2, 1, 3	89	1.08	-261.45	-211.83
r, 3, 3, 1, 3	126	-0.94	-86.92	-81.35
r, 4, 2, 2, 3	64	13.23	69.37	70.57
r, 3, 2, 1, 4	74	3.86	-16.10	-8.16

Table 2: The results with inequalities (14) for $\alpha = Q_2$ and $P^{'} = \emptyset$

problem name	no. of rows	% imp. in % root gap	% imp. in cpu	% imp. in nodes
e, 3, 2, 1, 3	66	4.91	26.18	45.20
e, 4, 2, 1, 3	86	3.23	28.98	29.13
e, 5, 2, 1, 3	109	2.75	35.27	41.65
e, 6, 2, 1, 3	131	6.48	37.76	45.35
e, 7, 2, 1, 3	140	6.02	55.75	59.92
e, 3, 2, 2, 3	66	3.09	39.39	46.90
e, 4, 2, 2, 3	86	1.17	46.40	52.36
e, 3, 3, 1, 3	121	-2.83	30.21	-1.90
e, 4, 3, 1, 3	169	1.15	20.05	27.61
e, 5, 3, 1, 3	221	2.62	18.79	29.55
e, 6, 3, 1, 3	243	4.74	-42.48	10.68
e, 7, 3, 1, 3	252	4.16	38.79	46.76
r, 5, 2, 1, 3	97	2.29	84.62	84.77
r, 3, 3, 1, 3	126	2.15	69.77	64.98
r, 4, 2, 2, 3	64	24.69	76.70	77.91
r, 3, 2, 1, 4	79	10.77	-22.14	-15.03

Table 3: The results with inequalities (14) for $\alpha = Q_2$ and $P^{'} = P_c$

problem name	no. of rows	% imp. in % root gap	% imp. in cpu	% imp. in nodes
e, 3, 2, 1, 3	147	-0.89	15.63	54.53
e, 4, 2, 1, 3	203	0.54	69.41	80.21
e, 5, 2, 1, 3	257	0.42	47.14	66.51
e, 6, 2, 1, 3	317	1.32	59.90	68.50
e, 7, 2, 1, 3	349	1.40	57.75	70.92
e, 3, 2, 2, 3	163	-1.50	-122.21	-32.44
e, 4, 2, 2, 3	228	-1.47	3.10	30.83
e, 3, 3, 1, 3	262	2.08	23.57	75.18
e, 4, 3, 1, 3	352	-5.26	39.80	66.04
e, 5, 3, 1, 3	449	4.42	75.36	94.56
e, 6, 3, 1, 3	514	7.27	87.86	94.74
e, 7, 3, 1, 3	546	7.69	87.01	95.38
r, 5, 2, 1, 3	288	1.08	23.02	49.83
r, 3, 3, 1, 3	335	-4.81	91.23	94.23
r, 4, 2, 2, 3	202	-3.11	-174.75	-108.19
r, 3, 2, 1, 4	261	1.44	36.21	55.13

Table 4: The results with inequalities (14) for $\alpha = kQ_2$ and $P^{'} = \{j \in P_c : q_{ij} = k\}$

Next, we repeat the same test for inequalities (16) and (18). Here we add all possible inequalities as their number is polynomial. The results are given in Tables 5 and 6.

Here, we observe that the first family of inequalities (16) improve the cpu time and the number of nodes for all problems except one. The average improvements in the percentage root gap, the cpu time and the number of nodes are 1.53%, 42.65% and 46.11%, respectively. For the second family of inequalities (18), we observe that the cpu time increased for 4 problems and the number of nodes increased for 3 problems. The average improvements in the percentage root gap, the cpu time, and the number of nodes are 0.55%, 25.70%, and 47.38%.

Based on these results, we decided to use the valid inequalities (14) with $\alpha = Q_2$ and $P^{'} = P_c$, the valid inequalities (14) with $\alpha = kQ_2$ and $P' = \{j \in P_c : q_{ij} = k\}$ and the valid inequalities (16). We report results with these three families of valid inequalities in Table 7.

These inequalities together decrease the root gap, the number of nodes in the branch and cut tree and the cpu time for all instances. The average, minimum, and maximum improvements in the cpu time are 59.1%, 14.3%, and 92.97%, respectively.

In the final part of our first experiment, we investigate the effect of our valid inequalities in solving other random instances (available at http://www.bilkent.edu.tr/~alpersen /Mixed_Pallet) with larger number of customers or periods. The results are tabulated in Tables 8 and 9. For most problem instances, the solver terminated running out of memory. We report, for both cases, with and without valid inequalities, the size of the formulations,

problem name	no. of rows	% imp. in % root gap	% imp. in cpu	% imp. in nodes
e, 3, 2, 1, 3	40	2.17	25.27	49.93
e, 4, 2, 1, 3	53	2.84	77.27	77.95
e, 5, 2, 1, 3	66	1.40	27.66	37.07
e, 6, 2, 1, 3	79	0.48	58.42	57.96
e, 7, 2, 1, 3	83	0.54	51.22	51.01
e, 3, 2, 2, 3	40	-2.38	-67.12	-49.65
e, 4, 2, 2, 3	53	-4.05	44.12	44.92
e, 3, 3, 1, 3	88	-1.37	7.22	18.16
e, 4, 3, 1, 3	124	-0.34	45.98	42.77
e, 5, 3, 1, 3	161	-2.94	56.01	52.05
e, 6, 3, 1, 3	174	3.42	42.74	44.20
e, 7, 3, 1, 3	178	6.11	73.11	73.92
r, 5, 2, 1, 3	57	1.31	72.62	75.52
r, 3, 3, 1, 3	88	4.47	93.89	94.99
r, 4, 2, 2, 3	35	13.18	71.57	64.52
r, 3, 2, 1, 4	36	-0.28	2.44	2.43

Table 5: The results with inequalities (16)

problem name	no. of rows	% imp. in % root gap	% imp. in cpu	% imp. in nodes
e, 3, 2, 1, 3	98	-0.36	13.82	44.61
e, 4, 2, 1, 3	133	0.48	59.13	67.53
e, 5, 2, 1, 3	166	-0.53	26.50	44.05
e, 6, 2, 1, 3	198	-0.45	-11.09	13.62
e, 7, 2, 1, 3	216	0.03	-36.36	-6.92
e, 3, 2, 2, 3	114	0.00	6.38	39.02
e, 4, 2, 2, 3	157	0.03	-48.10	-11.13
e, 3, 3, 1, 3	188	5.14	35.85	67.96
e, 4, 3, 1, 3	253	-6.87	48.58	70.05
e, 5, 3, 1, 3	322	0.39	61.04	78.75
e, 6, 3, 1, 3	358	3.58	73.88	85.43
e, 7, 3, 1, 3	376	7.13	80.45	89.21
r, 5, 2, 1, 3	184	1.08	43.31	57.20
r, 3, 3, 1, 3	216	-4.28	82.24	86.19
r, 4, 2, 2, 3	125	6.28	37.52	47.67
r, 3, 2, 1, 4	145	-2.87	-61.91	-15.15

Table 6: The results with inequalities (18)

problem name	no. of rows	% root gap	cpu (in sec.s)	no. of nodes	% imp. in % root gap	% imp. in cpu	% imp. in nodes
e, 3, 2, 1, 3	179	61.57	4.39	3469	1.85	20.37	69.17
e, 4, 2, 1, 3	244	66.46	46.15	52432	2.43	76.52	85.98
e, 5, 2, 1, 3	66	63.93	79.62	79096	1.40	68.34	82.29
e, 6, 2, 1, 3	381	62.33	407.28	388649	4.93	73.42	83.64
e, 7, 2, 1, 3	413	58.07	299.01	283087	6.02	68.23	79.94
e, 3, 2, 2, 3	195	56.67	51.09	64666	3.64	14.30	56.67
e, 4, 2, 2, 3	269	64.23	12982.43	15251520	2.36	33.52	57.41
e, 3, 3, 1, 3	303	53.99	9.83	2612	10.49	18.51	63.20
e, 4, 3, 1, 3	407	56.16	15.52	4265	8.59	24.02	63.81
e, 5, 3, 1, 3	522	55.84	41.94	12303	4.07	69.63	87.05
e, 6, 3, 1, 3	598	55.56	36.80	11903	10.44	87.23	94.04
e, 7, 3, 1, 3	630	52.46	33.81	7632	12.24	89.54	96.70
r, 5, 2, 1, 3	332	54.95	76.93	88723	2.62	85.69	91.57
r, 3, 3, 1, 3	381	69.99	155.04	140149	5.15	92.97	95.60
r, 4, 2, 2, 3	225	20.77	374.79	565135	24.69	73.52	81.66
r, 3, 2, 1, 4	299	41.93	128.59	171691	9.59	49.80	66.45

Table 7: The results with inequalities (14) with $\alpha = Q_2$ and $P^{'} = P_c$, inequalities (14) with $\alpha = kQ_2$ and $P' = \{j \in P_c : q_{ij} = k\}$ and inequalities (16)

.

the percentage root gap (computed using the best upper bound of the two cases), the cpu time, the best upper bound at termination and the remaining percentage gap (i.e., $\frac{ub-lb}{ub} * 100$ where ub is the final upper bound and lb is the final lower bound).

The final % gaps and the final upper bounds are smaller with valid inequalities for all of the instances except for one. For that single instance, the difference in final % gap is quite small. With valid inequalities, the solver could prove optimality for two instances, whereas without valid inequalities, the minimum final gap is 7.48%. The average final gap is 33.62% without valid inequalities and 24.16% with valid inequalities. These results show that the valid inequalities help compute better upper and lower bounds in general, but the problem for larger instances remains difficult to solve to optimality.

The second experiment tests the effect of using mixed pallets on total costs. For this experiment, we use two sets of instances. In the first set, there are two types of products and in the second set, the number of product types is three. In both sets, there are three periods and the number of customers goes from 3 to 7. The results are reported in tables 10 and 11. We solve the problem with no mixed pallets, one mixed pallet and two mixed pallets. For each variant, we report the optimal value. Let $cost_i$ be the optimal value for the problem with i mixed pallets, for $i = 0, 1, 2$. The quantities $\%imp_1$ and $\%imp_2$ are computed as $\frac{cost_1 - cost_0}{cost_0} * 100$ and $\frac{cost_2 - cost_1}{cost_1} * 100$.

problem name	no. of rows	no. of columns	% root gap	cpu (in sec.s)	best upper bound	final % gap
r, 3, 2, 1, 6	41	150	52.21	5118.74	591.7	16.52
r, 5, 2, 1, 4	48	153	39.27	4084.87	812.8	21.80
r, 4, 2, 2, 4	44	142	61.74	3100.62	605.6	59.21
r, 10, 2, 1, 3	92	261	64.82	2665.99	872	40.05
r, 8, 2, 1, 3	65	184	46.46	3152.60	757.9	26.73
r, 4, 3, 2, 3	91	285	75.30	2560.41	516.7	50.85
r, 6, 3, 2, 3	113	347	70.04	1955.40	798.3	47.37
r, 6, 2, 3, 3	51	145	41.90	2742.59	558.7	34.42
r, 4, 2, 1, 5	41	139	39.10	12953.72	663.6	7.48
r, 4, 2, 2, 5	41	139	38.71	3651.33	661.2	31.75

Table 8: Results without valid inequalities for randomly generated instances with larger number of customers or periods

problem name	no. of rows	% root gap	cpu (in sec.s)	best upper bound	final % gap
r, 3, 2, 1, 6	618	40.87	3364.17	591.7	0.00
r, 5, 2, 1, 4	449	37.40	6880.10	806.8	11.40
r, 4, 2, 2, 4	465	60.52	3485.45	600.8	57.46
r, 10, 2, 1, 3	613	64.38	3229.82	912.8	40.17
r, 8, 2, 1, 3	467	38.87	4812.57	757.9	17.16
r, 4, 3, 2, 3	472	60.34	3930.67	513.1	32.46
r, 6, 3, 2, 3	648	66.77	3559.15	793.5	27.20
r, 6, 2, 3, 3	401	39.93	3030.50	557.5	33.47
r, 4, 2, 1, 5	525	26.40	403.07	663.6	0.00
r, 4, 2, 2, 5	564	26.13	4877.77	661.2	22.28

Table 9: Results with valid inequalities for randomly generated instances with larger number of customers or periods

$\lvert C \rvert$	$cost_0$	$cost_1$	$\%imp_1$	$cost_2$	$\%imp_2$
3	410.5	271.9	33.76	226.3	16.77
4	576.4	359.8	37.58	311.8	13.34
5	699.7	437.5	37.47	342.7	21.67
6	809.7	499.5	38.31	404.7	18.98
7	840.5	530.3	36.91	435.5	17.88

Table 10: Results with 2 types of products

$\lvert C \rvert$	$cost_0$	$cost_1$	$\%imp_1$	$cost_2$	$\%imp_2$
3	565.5	321.9	43.08	278.1	13.61
4	795.15	405.45	49.01	352.65	13.02
5	1013.45	549.95	45.73	476.15	13.29
6	1123.45	659.95	41.26	538.15	18.46
7	1154.25	690.75	40.16	568.95	17.63

Table 11: Results with 3 types of products.

The results show that incorporating mixed pallets results in significant savings in inventory holding and backlogging costs for the beverage producer's customers. For all 10 instances, significant reductions in total cost are possible, even with the introduction of a single mixed pallet. Incorporating a second mixed pallet results in further savings for the customers, albeit with decreasing marginal returns. The message from this experiment is clear. For unpopular items, offering even a limited number of mixed pallets will lead to considerably lower costs than the case where the customers are allowed to order only in full pallets. Such a result will enable the beverage producer to operate with standard pallets (mixed or full) without having much impact on customer profitability and sales.

5. Conclusion

In this paper, we study a manufacturer that is designing standard mixed pallets for its various customers (that are differentiated by their demand mix) that cannot justify full pallet shipments for every product that they demand. We state the problem of the manufacturer as determining the designs of a given number of mixed pallets so as to minimize the total inventory holding and backlogging costs of its customers. First we show that the problem is NP–hard. We develop a mixed integer linear programming formulation and valid inequalities to strengthen the formulation. Our numerical study shows that the incorporation of mixed pallets improve the performance of customers considerably, even with restrictions and a limited number of mixed pallets. Our numerical investigation also shows that the valid inequalities help significantly in reducing the solution times, but the problem remains to be difficult for instances with higher dimensions. Therefore, one straightforward extension of our study would be the development and testing of heuristics. One may also consider the incorporation of manufacturer's own costs (such as inventory holding cost of pallets) to the model. Although the specific company that motivated this research works with customers with deterministic product demands, another logical extension is the introduction of probabilistic demands to the problem.

Acknowledgments

We are grateful to an anonymous referee for suggesting mixed integer rounding inequalities.

References

[1] Aichlmayr, M. 2002. Robots to the rescue. *Transportation & Distribution*, **43**, 9, page 60.

[2] Andel, T. 1998. Customer intimacy changes unit loads to rainbows. *Transportation & Distribution*, **39**, 8, page 104.

[3] Barrese, R. 2002. Outsourcing becomes ingredient to success: Pasta maker relies on 3PL to provide a variety of special services. *Food Logistics*, April 15, page 39.

[4] Belvaux, G. and L.A. Wolsey. 2000. *bc-prod*: A specialized branch-and-cut system for lot-sizing problems. *Management Science*, **46**, 724–738.

[5] Beverage Industry. 2001. The benefits of automation. **92**, 6, page 76.

[6] Bischoff, E.E. and M.S.W. Ratcliff. 1995. Loading multiple pallets. *Journal of the Operational Research Society*, **46**, 1322–1336.

[7] Food & Drug Packaging. 2000. Software. **64**, 10, page 71.

[8] Garey, M. R. and D. S. Johnson. 1979. *Computers and intractability: a guide to the theory of NP-Completeness*. Freeman, San Francisco.

[9] Gilmore, P. C. and R. E. Gomory. 1965. Multi stage cutting stock problems of two or more dimensions. *Management Science*, **13**, 94–119.

[10] Grocery Marketing. 1996. Squeezing out waste of space. **62**, 5, page 112.

[11] Hifi, M. Exact algorithms for unconstrained three-dimensional cutting problems: a comparative study. 2004. *Computers & Operations Research*, **31**, 657-674.

[12] Mazur, D.R. Integer programming approaches to a multi-facility location problem. 1999. PhD Thesis, John Hopkins University.

[13] Modern Materials Handling. 2001. Thirsty for profits. May 1, page 3.

[14] Marchand, H. and L.A. Wolsey. Aggregation and Mixed Integer Rounding to solve MIPs. 2001. *Operations Research*, **49**, 363–371.

[15] Miller, A.J. and L.A. Wolsey. Tight MIP formulations for multi-item discrete lot-sizing problems. 2003. *Operations Research*, **51**, 557–565.

[16] Nemhauser, G. and L.A. Wolsey. Integer and combinatorial optimization. Wiley, 1988.

[17] Pochet, Y. and L.A. Wolsey. Integer knapsack and flow covers with divisible coefficients: Polyhedra, optimization and separation. 1995. *Discrete Applied Mathematics*, **59**, 57-74.

[18] Pochet, Y. and L.A. Wolsey. Production Planning by Mixed Integer Programming. Springer, 2006.

[19] Redman, R. 1996. Loads of variety mixed pallets containing multiple varieties of popular products are delivering efficiencies to the store. *Supermarket News*, February 26, page 66.

[20] Witt, C. E. 1995. Transport packaging: the next frontier of cost reduction. *Material Handling Engineering*, **50**, 12, page 55.

[21] Yaman, H. The Integer Knapsack Cover Polyhedron. 2005. Technical Report, Bilkent University, Dept. of Industrial Engineering.